On the Escape Probability Estimation in Large Graphs

Konstantin Avrachenkov, Alexandra Borodina

On the Escape Probability Estimation in Large Graphs

Konstantin Avrachenkov
Inria, Sophia Antipolis
France
k.avrachenkov@sophia.inria.fr

Alexandra Borodina
Petrozavodsk State University,
Institute of Applied Mathematical Research
of the Karelian Research Centre of RAS
Petrozavodsk, Russia
borodina@krc.karelia.ru

***Abstract*—We consider the large graphs as the object of study and deal with the problem of escape probability estimation. Generally, the required characteristic cannot be calculated analytically and even numerically due to the complexity and large size of the investigation object. The purpose of this paper is to offer the effective method for estimating the probability that the random walk on graph first enters a node b before returning into starting node a. Regenerative properties of the random walk allow using an accelerated method for the cycles simulation based on the splitting technique. The results of numerical experiments confirm the advantages of the proposed method.**

I. Introduction

The rapid development of telecommunication networks and infocommunication systems makes the structure of the networks very complex. The models of random graphs are of great practical importance now but together with the growth of network structure, in turn, the models become more complex. So, the new research methods also become highly important for unreliability estimation or rare events simulation problems in random graphs and complex networks.

In this paper, we shall show how to estimate the escape probability. Generally, the need to estimate the escape probability or the *first entrance probability* arises for systems with failures and overflows. Typically those probabilities are small and traditionally estimated by splitting [1]. In our case, we are faced with a situation where the probability is not so small to count in this fashion, but the time on the standard random walk may be too large to get accurate estimates.

There is a lot more to this topic than our brief introduction suggests, but nowadays often accelerated techniques are used not only for evaluation in the context of rare events. The nice example of this can be seen in article [2], where the importance sampling principle is used for accurate estimation of the number of $s-t$ paths for general graphs and finding the expression for the expected number of such paths in a random graph. The other problems where the multilevel splitting approach becomes possible dealt with in [3]. There the graph coloring and the Potts model zero-temperature partition function approximation problem are solved by introducing an equivalent rare-event estimation problem. According to [4] another way of describing the continuous optimization problem gives the opportunity to apply the splitting method to it that is both very fast and accurate.

The outline of the paper is as follows. Firstly we describe the graph network model, define the escape probability to be estimated by the standard random walk. We discuss conditions that guarantee graph connectivity and formulate the algorithm for constructing a connected graph, which is equivalent to the original one in terms of the standard random walk.

In Section III we give methods for escape probability evaluation based on effective resistance [5], [6], [7] regenerative theory [8], [9] and rare-event simulation [1], [10]. We also propose to use the regenerative structure of Markov chain associated with the random walk in graph, which guarantees a strong consistency of the estimate for the desired probability. Then, we propose a modification of the standard splitting method [10] in order to accelerate random walk procedure in large graphs. Section IV shows the results of a comparative analysis of the methods considered in the paper. Using the splitting technique allows to speed up the random walk procedure and estimate with higher accuracy.

II. Preliminaries

Following the authors of [11], [12] we will consider undirected and connected (in view of the further remarks) graph $G=(V,E)$, with starting set $I_n \subset V$ of n initial nodes. Let $X=\{X_t,\ t\geq 0\}$ be a Markov chain associated with random walk in G with countable state space χ. The transition matrix of the standard random walk is

$$P=[p_{u,v}],\ u,v\in V,\ p_{u,v}=\alpha_{u,v}/d_u,$$

where d_u is the degree of node u and $\alpha_{u,v}$ is the number of edges between $u,v\in V$.

For some vertex $b\in V$ define the *escape probability* $p_{esc(a\to b)}$ that starting from a the random walk will reach b before returning to a. This probability can be formalized as follows

$$p_{esc(a\to b)}=\mathsf{P}\{X_j=b,\ X_i\neq a,\ 0\leq i\leq j<\infty|X_0=a\}.$$

We omit $(a\to b)$ whenever the pair a,b is clear from the context. This paper proposes to discuss existing ways to calculate p_{esc}. As a matter of fact, the computation of p_{esc} is a difficult problem for any analytical method, and it is necessary to apply simulation technics to estimate its value. This is natural and sometimes the only way to study the characteristics of large networks such as peer-to-peer or social networks.

The goal of the research is to speed-up the simulation for p_{esc} estimation in large graphs or in online social networks

where the network structure is not completely available. Besides, the results presented in [11], [12] give us reasons to estimate the escape probability by using the partial random walk based crawling. In this regard, one of the most important properties of graph G is connectivity. Indeed it seems that connectivity is an unrealistic requirement for real social web-graphs. Furthermore, connectivity is often a rare event in sparse graph and the special techniques are required for rare-event simulation of the probability that the graph is connected. In this regard, it is important to mention the paper [13] where the conditional Monte Carlo method is used for connectivity probability estimation for large random graphs.

If a graph G is not connected, then it is necessary to build a new corresponding graph $G^{'} = (V^{'}, E^{'})$ using the *super-node* instead of the initial set I_n. Moreover, the set I_n includes vertices from all different components of graph G. The following rules are used to build a connected graph $G^{'}$ that is equivalent to the original one with respect to a random walk in steady state:

1) merge all vertices of the set I_n into one super-node S_n;
2) build the set of vertices $V^{'} = \{V \backslash I_n\} \cup \{S_n\}$;
3) build the set of edges $E^{'} = E \backslash \{E \cap \{I_n \times V\}\} \cup \{(S_n, V) : \forall (u, v) \in E, u \in I_n, v \in V \backslash I_n\}$.

The advantages of this approach are described in the paper [12]. Based on this, without loss of generality, we will further consider the connected graph G and single starting node $a \in V$ instead of the set I_n.

Define the k-trajectory of a random walk as a sequence of visited nodes before returning to the starting node a

$$X_1^{(k)}, \ldots, X_{\xi_k}^{(k)},\ k \geq 1, \tag{1}$$

where the moments ξ_k are the first return times to a. The sequence $\{\xi_k\}_{k \geq 1}$ forms a renewal process, so the independent trajectories can be interpreted as regenerative cycles. Let π_i be the stationary distribution at node i in the random walk on G, then the following holds

$$\mathbb{E}[\xi_k] = 1/\pi_a,\ \pi_a = d_a/2|E|.$$

The connectivity of the graph G ensures that $\mathbb{E}[\xi_k] < \infty$, then the regenerative process X with the finite mean cycle length is positive recurrent.

In the subsequent discussion, we will focus on the escape probability estimating by using the regenerative simulation and will accelerate the regenerative cycles construction by splitting of the trajectories.

III. Escape probability estimation

Let $G = (V, E)$ be the weighted undirected graph and for $a, b \in V$ the probability p_{esc} that starting in a the random walk reaches b before returning to a has to be found. This measure called *escape probability* [5], [6].

In fact, this term means the same as the rest, such as *the probability on the regeneration cycle* [15], or the *first entrance probability* [1] came from different areas of applied mathematics.

1) Effective resistance: There are analytical results in [5], [6], [7] based on relation between elementary electric network theory and random walk on undirected weighted graph G which is viewed as electrical network. Each edge $(i, j) \in E$ is a resistor with resistance $r_{ij} = 1/w_{ij}$, where w_{ij} is a weight of edge (i, j).

Thus, the escape probability is related to the effective resistance

$$p_{esc} = \frac{1}{d_a R_{ab}}, \tag{2}$$

where $d_a = \sum_v w_{av}$ is the weighted degree of node a.

As follows from [6], escape probability p_{esc} can by computed as a solution of a linear system. Let L be the Laplacian matrix $n \times n$ of the graph G:

$$L_{ij} = \begin{cases} d_i = \sum_{k=1}^{n} w_{ik} & \text{if } i = j, \\ -w_{ij} & \text{otherwise.} \end{cases} \tag{3}$$

Equivalently the Laplacian can also be defined as $L = D - W$, where D is diagonal matrix with elements $D_{ii} = \sum_j w_{ij}$, W is weight matrix with elements $W_{ij} = w_{ij}$. Then consistently applying the Kirchhof's current law and Ohm's law the following matrix equation will be obtained

$$Lv = I_{ab}(e_a - e_b), \tag{4}$$

where v is the vector of potentials, I_{ab} is the current from a to b, the vector e_i is the standard basis vector of $\mathbb{R}^n$ with value one on the i-th position and zeroes on the others. From (4), it follows that

$$R_{ab} = (v_a - v_b)/I_{ab}, \tag{5}$$

where the current from a to b can be fixed $I_{ab} = 1$, so $R_{ab} = v_a - v_b$.

It is typically impossible to solve analytically the system of equations for large graph, but in some special cases this can be done numerically. For example, by using the python library linalg. However again, such an approach is problematic for large graphs because of matrices with large dimension and operations with them. Sometimes it is even slower than simulation.

2) Regenerative simulation and standard Monte-Carlo: It is well known that the irreducible and recurrent Markov chain is regenerative with the moments where the chain returns to any state, namely state a in our interest. Since a connected graph is considered, it is guaranteed that $\mathbb{E}[\xi_k] < \infty$ and the regenerative simulation method can be used (see e. a. [9], [14]).

In general, the regenerative simulation method based on the advantage of the regenerative structure of the process. Since the cycles and cycle lengths are independent and identically distributed (i.i.d.), then their time-average properties are consequence of the Central Limit Theorem and Renewal-Reward theorem [8], [15]. For the sake of completeness, it will be correct to give a general description of the regenerative method [15].

Consider the classical regenerative process $X = \{X(t),\ t \geq 0\}$ with regeneration moments $T_0 = 0 < T_1 < \ldots$ and i.i.d. cycle lengths $\{\alpha_i\}_{i \geq 1}$. Let f be any measurable function, then for i-th cycle calculate the value of function

f over the cycle Y_i and denote the sequence of i.i.d. random variables (r.v.)

$$Y_i = \int_{T_{i-1}}^{T_i} f(X(t))dt, i \geq 1. \quad (6)$$

So, an unknown stationary characteristic of the process X may be found as a ratio of expectations

$$\gamma = \mathbb{E}f(X) = \frac{\mathbb{E}Y_1}{\mathbb{E}\alpha_1}.$$

If $\mathbb{E}|Y_1| < \infty$ and $\mathbb{E}\alpha_1 < \infty$ then for regenerative estimator $\hat{\gamma}$ over n cycles holds w. p. 1

$$\widehat{\gamma} = \frac{n^{-1}\sum_{k=1}^{n} Y_k}{n^{-1}\sum_{k=1}^{n} \alpha_k} := \frac{\overline{Y_n}}{\overline{\alpha_n}} \to \gamma \; as \; n \to \infty. \quad (7)$$

Now let us run n independent trajectories of a random walk, starting from vertex a in a graph G. In the case of estimating the probability p_{esc} regenerative estimator (7) is greatly simplified. Actually, for every regeneration cycle denote the indicator $I^{(i)} = 1$ if the random walk visited the vertex b on the i-th cycle (i.e., before returning to vertex a). When we estimate the escape probability, then we have a degenerate situation because the whole cycle is represented by the only indicator and the cycle length is irrelevant (is equal to one in (7)). Thus, the target estimator (7) is reduced to the standard Monte-Carlo form:

$$\widehat{p}_{esc}[MC] = \frac{1}{n}\sum_{i=1}^{n} I^{(i)}. \quad (8)$$

If the trajectory ends after returning to the start node a, then the cycles will be built entirely. But it should be noted that when we deal with first-entrance probabilities it is not necessary to build a full cycle if the achievement of b has already happened. This trick can significantly save simulation time.

On the other hand, the regenerative interpretation allows offering an additional way for p_{esc} estimation. Namely, now let N be the number of cycles before (and including) the cycle with hitting b. Since the transition to the node b on the cycle occurs with probability p_{esc} then the r. v. N has a geometric distribution:

$$\mathsf{P}(N = k) = (1 - p_{esc})^{k-1} p_{esc}, \; k \geq 1.$$

Thus, an alternative estimator $\widehat{p}_{esc}[EN]$ can be immediately calculated from the standard estimate for expectation

$$\mathbb{E}[N] = 1/p_{esc}. \quad (9)$$

Disadvantages of Monte-Carlo estimator generally discussed in the context of of rare event probabilitiy estimation (for instance see [10], [16], [17], [18]). Recall that if $p_{esc} \to 0$, then the *relative error*

$$RE[\widehat{p}_{esc}] = \frac{\sqrt{Var[\widehat{p}_{esc}]}}{\mathbb{E}[\widehat{p}_{esc}]} \sim \frac{1}{\sqrt{np_{esc}}} \to \infty,$$

where $x \sim y$ means that $x/y \to 1$. So, for example, if $p_{esc} \approx 10^{-12}$ then to guarantee modest 10% RE the number n is about 10^{14} of experiments is required. Thus, the simulation time with a given accuracy can be unacceptably long. In practice, this means that the Monte-Carlo method gives a zero point estimate.

There are several rare event simulation techniques [17], [18] which allow reducing the variance of estimator or make a rare event less rare [10], [16] due to splitting technique.

3) The multilevel splitting: The need to use accelerated simulation methods occurs not only in the case of rare events. In particular, for large graphs, the probability p_{esc} may not be very small, however, the time for a random walk in a large graph may be unacceptably long. We propose the following splitting algorithm to speed-up the simulation of the random walk trajectories. We will use the regenerative structure of Markov chain $\{X_t, \; t \geq 0\}$ associated with random walk in a graph G and will apply the regenerative splitting method [19] for p_{esc} estimation. For more details about splitting method see [20], [21], [22], [23].

The main idea for rare events is to replace the estimation of a very small quantity by the estimation of several, not so small quantities, whose product is equal to the desired probability. For this purpose the system of thresholds is introduced. The trajectories multiply at each threshold and a branching process is formed. The paths that reach the next threshold are called successful. Thus, by reproducing only successful trajectories we artificially increase the probability of reaching the "rare region". This artificial acceleration must be compensated by the special factor in the final estimator. We define the key points for implementing the splitting method for random walk in large graphs. Visualization of the approach is shown in Fig. 1.

Initial state. The chain starts from node a; b is a target node (or the set of nodes).

Importance function. Define the importance function $S : \chi \to \mathbb{R}$, where $S(x)$ is the shortest distance between current x and final b given by Dijkstra's algorithm. We assume that the vertex x belongs to the "rare region" B if $S(x) = 0$, so

$$B = \{x \in \chi : S(x) = 0\}.$$

Let $l := S(a)$ be the shortest distance from the starting vertex a to the final b (or to the final set).

Find a sequence of level sets

$$\chi_0 \supset \chi_1 \supset \cdots \supset \chi_M,$$

where $\chi_k = \{x \in \chi : \; S(x) \leq l_k\}$ is a level sets of function S for levels $l_i \in [0, l]$, $i \in [0, M]$

$$l = l_0 \geq l_1 \geq \cdots \geq l_M = 0.$$

Note that unlike the standard splitting algorithm, where the entire process state space is divided into nested subsets, in our case, not necessarily $\chi_0 = \chi$ because there are no vertices with the distance $S(x)$ greater than l in consideration. However, a random walk can reach such vertices.

For $k = 1, \ldots, M$ define the nearest moment of hitting the level l_k:

$$T_k = \inf\{t > 0 : S(X_t) \leq l_k\}.$$

Then we define the sequence of nested events:

$$E_M \subset E_{M-1} \subset \cdots \subset E_0,$$

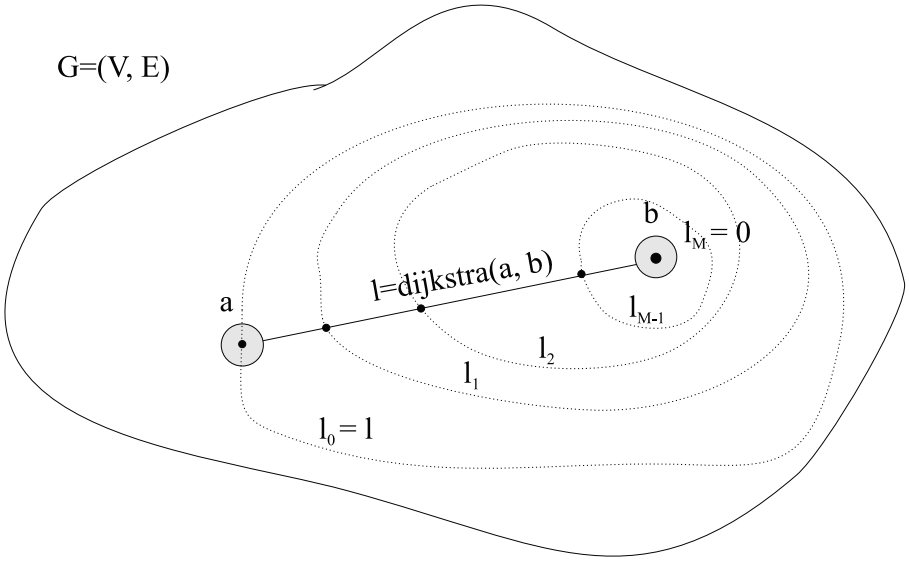

Fig. 1. Level sets determined by the Dijkstra algorithm

where $E_k = \{T_k < \tau_A\}$, $\tau_A = \inf\{t \geq 0 : X_{t-1} \notin A \,\&\, X_t \in A\}$, $A = \{x \in \chi : S(x) = l\}$. A set A can consist of one vertex a or several vertices if required. The fact that event E_k occurred means that intersection of the k-th threshold happened before the returning to the initial state a.

Transition probabilities p_k are the conditional probabilities that the trajectory hits the k-th level provided that the previous $(k-1)$-th level was reached

$$p_k = \mathbb{P}[E_k | E_{k-1}],\ k = 1, \ldots, M.$$

Since the events E_k are nested, the fact that the trajectory reached the level of k means that it crossed all previous levels $1, \ldots, k-1$ automatically. The quantity of interest p_{esc} is given by the telescopic product

$$p_{esc} = \prod_{k=1}^{M} p_k = \mathbb{P}[E_M]. \tag{10}$$

Splitting estimator. Denote r_k the number of splits at the level l_k

$$\widehat{p}_{esc}[SP] = \prod_{k=1}^{M} \widehat{p}_k, \quad \widehat{p}_k = \frac{N_k}{r_{k-1} N_{k-1}}, \tag{11}$$

where N_k is the number of hittings the level k and $r_{k-1}N_{k-1}$ is the total number of trajectories, starting from level $k-1$. Sometimes it is more convenient to use the final formula at once

$$\widehat{p}_{esc}[SP] = \frac{N_M}{r_0 r_1 \cdots r_{M-1}}, \tag{12}$$

where N_M is the number of rare events (hittings the threshold M). Another way to describe the splitting estimator (12) is the regenerative approach given by (6) and (7).

As it was for the naive Monte-Carlo method, denote the indicator $I^{(i)} = 1$ if the random walk visited the vertex b on the i-th cycle. Again it means that each cycle is represented by the value of a single indicator $I^{(i)}$. Then, at each threshold i we generate r_i starts of the random walk. So, each original path starting form a generates $D = r_1 \cdots r_{M-1}$ dependent subpaths with the same pre-history (dependable regenerative cycles). The total number of cycles is

$$n = r_0 \cdots r_{M-1} = r_0 D. \tag{13}$$

The *groups of cycles* corresponding to different starts from a are independent of each other and the cycles belonging to different groups are independent by construction. The total number of groups is denoted by r_0. From (6) for every i-th group denote

$$Y_i = \sum_{j=(i-1)\cdot D+1}^{i \cdot D} I^{(j)}, i = 1, \ldots, r_0,$$

where indicator $I^{(j)} = 1$ for the cycle with hitting b and $I^{(j)} = 0$ otherwise. All Y_i are i.i.d. From the regenerative properties of the sequence $\{I^{(j)}, j \geq 1\}$ with p. 1 holds:

$$\widehat{p}_{esc}[SP] = \frac{\sum_{j=1}^{r_0} Y_j}{r_0 \cdot D} \to \frac{\mathbb{E} \sum_{j=1}^{D} I^{(j)}}{D} = p_{esc}, r_0 \to \infty. \tag{14}$$

Further, we will use formula (14) for multilevel splitting simulation in graph Enron.

Stopping conditions. To stop the random walk trajectories, several variants of moments are potentially possible:

1) when the trajectory reaches the set A, i.e. returns to a. In this case, completed regeneration cycles are simulated;
2) when the E_M event occurred. This case may be justified for escape type point probabilities, when the trajectory can be truncated after the desired event has occurred;
3) after a random time (for example, exponentially distributed), if the hitting in the "rare region" didn't occur;

TABLE I. COMPARISON OF THE METHODS: ER VS. REGENERATIVE MC AND EN

N	$a \to b$ (d)	$\widehat{p}_{esc}[ER]$	t_{ER} s.	$\widehat{p}_{esc}[MC]$	t_{MC} s.	$\widehat{p}_{esc}[EN]$	t_{EN} s.
10	$0 \to 5(2)$	0.526	0.003	0.526	0.438	0.527	0.241
100	$0 \to 5(2)$	0.494	0.0009	0.493	5.14	0.492	5.29
1000	$0 \to 5(2)$	0.488	0.15	0.490	102.3	0.487	106.9
10000	$0 \to 5(2)$	0.501	135.3	0.501	697.5	0.501	647.2
20000	$0 \to 5(3)$	0.379	1793.4	0.368	548.1	0.369	466.2
40000	$6 \to 11(4)$	0.353	3601.8	0.350	2826.0	0.374	856.4
40000	$1 \to 6(8)$	0.367	4028.2	0.368	1036.1	0.367	781.0
100000	$0 \to 5(6)$	–	> 5000	0.573	2773.1	0.573	2442.0

4) when the trajectory has moved far enough away from the threshold from which it started.

The choice of a specific variant strongly depends on the problem and type of target probability.

Optimal parameters. The main question is how to choose the **optimal** levels $\{l_i\}$ and splitting factors $\{r_i\}$. For standard splitting, an optimal distance between thresholds and an optimal number of split paths at each threshold are defined by pilot run [10], [24]. Pilot run gives biased estimator but defines threshold partition in accordance with the requirement: p_i is not a rare event probabilities. In addition, it is considered optimal to choose such parameters when the number of branching trajectories does not grow exponentially on the one hand and the process is not damped on the other. For the standard splitting procedure the optimal values are related by the ratio

$$r_i = 1/p_i.$$

To clarify the splitting procedure, an *adaptive algorithm* is usually used that allows selecting parameters for the current process based on the conditions described above, but it gives a biased estimate of the probability. The key points of adaptive algorithm [24] are:

1) to get rough estimates of p_i by Monte-Carlo initializing the number of starts r_0 from l_0 arbitrarily;
2) to get splitting factors $\widehat{r}_i$ as random integer values with expected value $1/\widehat{p}_i$

$$\widehat{r}_i \sim \mathsf{Ber}(\frac{1}{\widehat{p}_i} - \left\lfloor \frac{1}{\widehat{p}_i} \right\rfloor) + \left\lfloor \frac{1}{\widehat{p}_i} \right\rfloor$$

In the general splitting algorithm, it is sometimes possible to select and set the optimal levels themselves. For our purpose, the thresholds $\{l_i\}$ are fixed and defined by S.

It is well known that social network graphs are sparse with a small diameter ("small world property"), then the number of thresholds practically is about $M \approx 12$ or can be even smaller. This is what motivates the use of the splitting method for realizing random walk in such graphs.

IV. NUMERICAL RESULTS

All numerical tests were executed on ultrabook HP ENVY Intel(R) Core(TM) i3 7100U 2.4GHz processor with 4GB of RAM, running Windows 10. All scripts were written in Python 3 using a library NetworkX and standard math modules.

1) Erdos-Renyi random graphs: Using the built-in graph generator from the NetworkX library we get graphs $G(N, p)$ according to the Erdos-Renyi model of different sizes and compare three ways to calculate the escape probability:

ER due to effective resistance formulae (2);
MC by simulation of regeneration cycles via Monte-Carlo method (8) *without truncation* after hitting b on the cycle (1st stopping condition);
EN by estimating the expectation $\mathbb{E}[N]$ (see (9)) over regeneration cycles *with truncation* after hitting b (2nd stopping condition).

In all cases we get the random graph from generator with parameters (N, p), where N is the number of nodes, p is the edge probability; then save them and running the methods on the same data. Each MC and EN estimate in the Table I was derived as a sample mean using 100 runs with $n = 10000$ cycles per run. For each pair of vertices in the Table I, the Dijkstra distance d is indicated in brackets.

In fact, the methods MC and EN are equivalent in terms of the simulation time on the same number of cycles. They can be used together to verify the correctness of the algorithm. For this purpose, it is necessary that stopping conditions be the same. But in our testing examples, we stop the cycle after hitting b in EN, and build the cycle completely in MC. As to be expected, the results of the probability calculations are the same for both methods, but you can see in the Table I how much time we need to lose in order to build complete cycles. The results presented in the Table I show that it is possible to use the truncation of the insignificant part of the cycle after hitting b for point estimation (but not for regenerative confidence estimation in general). The tests show that the ER method essentially loses in the rate of simulation with if the number of vertices in the graph increases.

2) Enron results: Next, consider the graph Enron representing the email communication network that covers all the email communication within a dataset of around half million emails. Nodes of the network are email addresses and if an address i sent at least one email to address j, the graph contains an undirected edge from i to j. The Enron data are available in an open database of Stanford university http://snap.stanford.edu/data/email-Enron.htm

Enron is undirected graph with $|V| = 36692$ number of nodes and $|E| = 183831$ edges. The giant component contains 33696 vertices and 180811 edges. According to SNAP statistics the diameter is equal to 11. However, this is not entirely correct, since there are vertices with eccentricities equal to 12 and 13.

For Enron we run MC and EN algorithms with the same stopping conditions as for the previous example but using 100 runs with n cycles per run. For the most time-consuming case, when the whole cycle is built (1st stopping condition), we apply the splitting algorithm SP and the estimator (14). This method should be compared with the MC method, since both

methods build the whole cycle without truncations after hitting b. Since the comparison must be fair, we use the same number n of cycles in both methods. In MC method, we start from the node a n times. In SP method the total number of cycles depends on the number of splits at each level and is calculated by the formula (13), where estimates $\widehat{r}_i$ are used instead of values r_i.

As noted above, we are not talking about rare events, but the small diameter of the Enron graph allows us to accelerate the random walk due to the splitting technique.

The experimental results show ambiguity in the applicability of the EN method for escape probability estimation in Enron. Thus, for some pairs of vertices, the estimate based on a system of linear equations doesn't match with the estimates obtained by the other methods. More likely this is caused by ill-conditioning of the linear system method. Further, we will illustrate the situations when the estimates coincide and when the method ER fails.

Example 1. $100 \to 25000$ (d=5). In this case, all methods give a similar value for escape probability. The numerical results are presented in the table II. All estimates except the ER method are counted on the same number of cycles n.

TABLE II. COMPARISON OF THE METHODS: ER VS. REGENERATIVE MC, EN, SP

Method	n	$\widehat{p}_{esc}$	time s.
ER	-	0.198	235.5
EN	1000	0.206	800.2
MC	1000	0.183	861.6
SP	1000	0.184	692.2

The splitting method SP gives a small time advantage over the naive Monte-Carlo. The estimates of the optimal splitting numbers $\widehat{r}_i$ were received from the pilot run.

Example 2. $8555 \to 27718$ (d=13). Consider two vertices at the maximum distance from each other $d = 13$. For this example results from the Table III show that the method ER gives an inadequate value for $\widehat{p}_{esc}[ER] = 0.8$ while other estimates are close. This behavior was also observed for vertices not necessarily strongly distant from each other

TABLE III. COMPARISON OF THE METHODS: ER VS. REGENERATIVE MC, EN, SP

Method	n	$\widehat{p}_{esc}$	time s.
ER	-	0.8007	161.1
EN	100	0.094	83.4
	1000	0.131	784.2
	2000	0.136	1646.1
MC	100	0.145	185.0
	1000	0.129	1325.7
	2000	0.136	3028.7
SP	1024	0.138	998.8
	2000	0.142	1784,3
SP over	1024	0.068	375.9
	2000	0.038	417.7

As in the previous example, the splitting method SP builds the estimate faster than the MC method. In total, such acceleration will significantly reduce the time for confidence estimation. In the Table III are also presented results for the case when the retrials numbers $\widehat{r}_i$ are not optimal. Let now the retrials chosen so, that the splitting occurs predominantly at the upper thresholds. Under these conditions, the expected over-brunching occurs and the estimator becomes distorted as we see in the line named "SP over" in the Table III.

V. CONCLUSION

In this paper, we reviewed the possible methods for escape probability estimating and investigated their applicability for large graphs. The numerical results show the advantage of the regenerative splitting approach which we have proposed to speed up the random walk. As it turned out, this approach gives an improvement even in the case when the estimated values are not very small. Further analysis is needed to develop this method for the networks with attracting nodes, where the escape probability may be really small.

ACKNOWLEDGMENT

The study was carried out under state order to the Karelian Research Centre of the Russian Academy of Sciences (Institute of Applied Mathematical Research KarRC RAS) and supported by the Russian Foundation for Basic Research, projects 18-07-00187, 18-07-00147.

REFERENCES

[1] T. Dean, P. Dupuis, "Splitting for rare event simulation: A large deviation approach to design and analysis", *Stochastic processes and their applications*, vol. 119, no. 2, 2009, pp. 562-587.

[2] B. Roberts, D. P. Kroese,"Estimating the Number of st Paths in a Graph", *Journal of Graph Algorithms Applications*, vol. 11, no. 1, 2007, pp. 195-214.

[3] R. Vaisman, M. Roughan, D. P. Kroese,"The Multilevel Splitting algorithm for graph colouring with application to the Potts model", *Philosophical Magazine*, Vol.97, no.19, 2017, pp. 1646-1673.

[4] Q. Duan, D. P. Kroese, "Splitting for optimization", *Computers & Operations Research*, vol. 73, 2016, pp. 119-131.

[5] P. G. Doyle, J. L. Snell, "Random walks and electric networks", *arXiv preprint math/0001057*, 2000.

[6] V. S. S. Vos, *Methods for determining the effective resistance*. Masters thesis, 20 December 2016.

[7] W. Ellens, F. M. Spieksma, P. Van Mieghem, A. Jamakovic, R. E. Kooij. "Effective graph resistance", *Linear algebra and its applications*, Vol. 435., No. 10, 2011, pp. 2491-2506.

[8] S. Asmussen, *Applied probability and queues*. Vol. 51. Springer Science & Business Media, 2008.

[9] P. W. Glynn, "Some topics in regenerative steady-state simulation", *Acta Applicandae Mathematica*, vol. 34, no. 1-2, 1994, pp. 225-236.

[10] R. Y. Rubinstein, D. P. Kroese, *Simulation and the Monte Carlo method*. New Jersey: John Wiley & Sons, Inc., 2017

[11] K. Avrachenkov, B. Ribeiro, D. Towsley, "Improving Random Walk Estimation Accuracy with Uniform Restarts", *in Proceedings of WAW 2010*, Also LNCS vol. 6516, December 2010, pp. 98-109.

[12] K. Avrachenkov, B. Ribeiro, J. K. Sreedharan, "Inference in OSNs via Lightweight Partial Crawls", *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, 2016, pp. 165-177.

[13] S. Rohan, C. Hirsch, D. P. Kroese, V. Schmidt, "Rare event probability estimation for connectivity of large random graphs" *Proceedings of the 2014 Winter Simulation Conference*, IEEE Press, 2014, pp. 510-521.

[14] P. W. Glynn, D. L. Iglehart."Conditions for the applicability of the regenerative method", *Management Science*, vol. 39, no. 9, 1993, pp. 1108-1111.

[15] K. Sigman, R. W. Wolff, "A review of regenerative processes", *SIAM review*, vol. 35, no. 2, 1993, pp. 269-288.

[16] R. Y. Rubinstein, A. Ridder, R. Vaisman, *Fast Sequential Monte Carlo Methods for Counting and Optimization*. New Jersey: John Wiley& Sons, Inc., 2014.

[17] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods*. John Wiley & Sons, 2011.

[18] S. M. Ross, *Simulation*. Elsevier, 2006.

[19] A. V. Borodina, ”PhD Thesis. Regenerative modifcation of the splitting method for estimating the overload probability in queuing systems”, Petrozavodsk State University, 2008. (in russian)

[20] M. Garvels, ”PhD Thesis. The splitting method in rare event simulation”, The University of Twente, The Netherlands May, 2000.

[21] P. Glasserman, P. Heidelberger, P. Shahabuddin, T. Zajic, ”Splitting for rare event simulation: analysis of simple cases”, *Proceedings of the 1996 Winter Simulation Conference*, 1996, pp. 302–308.

[22] P. E. Heegaard, ”A survey of Speedup simulation techniques”, *Workshop tutorial on Rare Event Simulation*, Aachen, Germany, 1997.

[23] P. Heidelberger, ”Fast simulation of rare events in queuieng and relaibility models”, *Performance Evaluation of Computers and Communications Systems*, Springer-Verlag, LN in Computer Sci, vol. 729, 1993, pp. 165–202.

[24] Z. I. Botev, D. P. Kroese, ”Efficient Monte Carlo simulation via the generalized splitting method”, *Statistics and Computing*, vol. 22, no. 1., 2012, pp. 1-16.